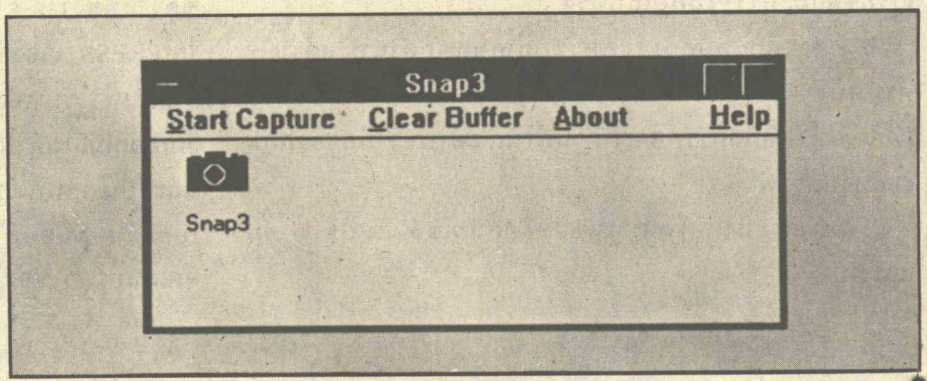
เอนก ศิรจริยาภิวัฒน์

เก็บภาพบนวินโดวส์ด้วย SNAP3

แนะนำยูทิลิตี้ที่ช่วยในการเก็บภาพบน Windows

มุมยูทิลิตี้ที่เสนอท่านผู้อ่านมาตลอดในช่วงนี้ จะสังเกตได้ว่าเป็นยูทิลิตี้เกี่ยวกับจอภาพ ในคราวนี้ผมเองได้เห็นบทความเกี่ยวกับโปรแกรมเก็บภาพบนวินโดวส์ในวารสาร Dr. Dobb's Journal จึงนำมาฝากเพื่อเป็นแนวทางสำหรับการพัฒนาโปรแกรมในลักษณะเดียวกันนี้ สำหรับโปรแกรม SNAP นั้นเดิมทีเป็นโปรแกรมสาธารณะที่อยู่บน Compuserve แต่เขียนสำหรับไมโครซอฟต์วินโดวส์เวอร์ชัน 2.0 ทำให้เมื่อเรียกโปรแกรมนี้บนไมโครซอฟต์วินโดวส์ เวอร์ชัน 3.0 หรือ 386 Enhanced mode จึงมีข้อความเตือนขึ้นมาตลอด โปรแกรมนี้จึงถูกพัฒนาขึ้นใหม่โดย จิม คอนเกอร์ เมื่อเราเรียกโปรแกรม SNAP3 เราจะเห็นเมนูดังรูปที่ 1 โดยเมนูแรกจะเป็น Start capture เมื่อเราเลือกเมนูนี้ เราจะเห็นเมาส์เคอร์เซอร์เป็นรูปกากบาท (crosshair) โดยเราเลื่อนรูปกากบาทนี้ไปด้านซ้ายบนเพื่อกำหนดจุดเริ่มต้นขอบเขตของรูปที่จะเก็บ เสร็จแล้วกดปุ่มซ้ายของเมาส์ค้างไว้ลากมาตำแหน่งขวาล่างเพื่อกำหนดขอบเขตที่เก็บรูปภาพแล้วปล่อยปุ่มที่กดไว้ รูปภาพที่จะเก็บจะไปปรากฏในวินโดว์ของ Snap3 สำหรับอีกเมนูคือ Clear Buffer จะเป็นการเคลียร์รูปภาพที่อยู่ในวินโดว์ของ Snap3 และคลิปบอร์ด ด้วยโปรแกรม Snap3 นี้เราจะสามารถเก็บรูปภาพบางส่วนที่อยู่ในวินโดว์ได้ ซึ่ง Snap3 จะส่งผ่านรูปที่เก็บไปให้คลิป-



รูปที่ 1 แสดงเมนูของ Snap 3

```
ALL: snap3.exe

snap3.obj : snap3.c
     cl -AS -c -DLINT_ARGS -Gsw -Oat -W2 -Zped snap3.c

snap3.res:  snap3.rc snap3.ico
     rc -r snap3.rc

snap3.exe : snap3.obj snap3.def snap3.res
     link /NOD snap3, , ,libw slibcew, snap3.def
     rc snap3.res
```

รูปที่ 2 ส่วนของโปรแกรม make file

```
NAME          SNAP3
DESCRIPTION   'snap3 program for windows bitmap capture to clipboard'
EXETYPE        WINDOWS
STUB          'WINSTUB.EXE'
CODE          PRELOAD MOVEABLE
DATA          PRELOAD MOVEABLE MULTIPLE
HEAPSIZE      1024
STACKSIZE     4096
EXPORTS       WndProc
```

รูปที่ 3 ส่วนของนิยาม (defination)

```
/* snap3.rc */
#include "snap3.h"

snap3   ICON    snap3.ico
snap3 MENU
BEGIN
        MENUITEM "&Start Capture"       IDM_START
        MENUITEM "&Clear Buffer",       IDM_CLEAR
        MENUITEM "&About",                      IDM_ABOUT
        MENUITEM "\a&Help",                     IDM_HELP
END
```

รูปที่ 4 ส่วนของโปรแกรม resource file

```
/* snap3.h */

#define IDM_START   1   /* menu item id values */
#define IDM_CLEAR   2
#define IDM_ABOUT   3
#define IDM_HELP    4

/* function prototypes */
long  FAR PASCAL WndProc (HWND, unsigned, WORD, LONG) ;
void  OutlineBlock (HWND hWnd, POINT beg, POINT end) ;
```

รูปที่ 5 ส่วนของ header file

บอร์ด ซึ่งภาพที่อยู่คลิปบอร์ดนี้จะสามารถนำไปใช้ประโยชน์ต่าง ๆ ได้มากมาย เช่น นำไปทำคู่มือ ใช้ในเวิร์ดโปรเซสเซอร์ หรือจะนำไปพิมพ์หรือแปลงรูปแบบโดยพิซซาพลัสหรือไฮจาค เพื่อนำไปใช้ในโปรแกรมประยุกต์ต่าง ๆ ได้มากมาย

สำหรับซอร์สโค้ดจะเป็นไปตามรูปที่ 2 ถึง 6 ซึ่งอันแรกสุดในรูปที่ 2 จะเป็นข้อมูลสำหรับ NMAKE เพื่อเรียกตัวแปลภาษาซีของไมโครซอฟต์ (CL), resource compiler (RC) และ link เพื่อสร้างโปรแกรม Snap3 ให้เรา รูปที่ 3 จะเป็นส่วนให้นิยาม (defination) เช่น ชื่อโปรแกรม หรือการจัดการส่วนต่าง ๆ ซึ่งจะเก็บไว้ในไฟล์ชื่อ SNAP3.DEF รูปที่ 4 จะเป็นส่วนของ resource file ซึ่งจะบอกถึง program's icon (เราสามารถสร้างไอคอนจากไอคอนเอดิเตอร์) และกำหนด (define) เมนูต่าง ๆ รูปที่ 5 จะเป็นส่วนของไฟล์เฮดเดอร์ สำหรับตัวโปรแกรมจริง ๆ จะอยู่ในรูปที่ 6 ซึ่งเราจะสามารถสร้าง SNAP3 ได้นั้น เราจำเป็นต้องมี Software Development Toolkit (SDK) และไมโครซอฟต์ซี ซึ่งการทำ SNAP3 นี้คงไม่ยากเกินไป และหวังว่า SNAP3 จะเป็นประโยชน์กับท่านผู้อ่านมาก เพราะเราจะได้ส่วนของโปรแกรมเกี่ยวกับจอภาพได้ครบถ้วนตามที่เราต้องการ สำหรับยูทิลิตี้คราวนี้อาจจะสั้นไปสักหน่อย แต่สำหรับคำว่าประโยชน์คงไม่สั้นตามนะครับ

รูปที่ 6 โปรแกรมหลักของ Snap 3

```
/* snap3.C - Screen Capture to clipboard - jim conger  */

#include <windows.h>
#include <stdlib.h>
#include "snap3.h"

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpszCmdLine,
    int nCmdShow)
{
     static char szAppName [] = "snap3" ;
     HWND        hWnd ;
     MSG         msg ;
     WNDCLASS    wndclass ;

     if (!hPrevInstance)
     {
          wndclass.style         = CS_HREDRAW | CS_VREDRAW ;
          wndclass.lpfnWndProc   = WndProc ;
          wndclass.cbClsExtra    = 0 ;
          wndclass.cbWndExtra    = 0 ;
          wndclass.hInstance     = hInstance ;
          wndclass.hIcon         = LoadIcon (NULL, szAppName) ;
          wndclass.hCursor       = LoadCursor (NULL, IDC_ARROW) ;
          wndclass.hbrBackground = GetStockObject (WHITE_BRUSH) ;
          wndclass.lpszMenuName  = szAppName ;
          wndclass.lpszClassName = szAppName ;

          if (!RegisterClass (&wndclass))
               return FALSE ;
     }
     hWnd = CreateWindow (szAppName, "Snap3",
          WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT,
          GetSystemMetrics (SM_CXSCREEN) / 2,
          8 * GetSystemMetrics (SM_CYMENU),
          NULL, NULL, hInstance, NULL) ;
     ShowWindow (hWnd, nCmdShow) ;
     UpdateWindow (hWnd) ;

     while (GetMessage (&msg, NULL, 0, 0))
     {
          TranslateMessage (&msg) ;
          DispatchMessage (&msg) ;
```

รูปที่ 6 (ต่อ)

```
    }
    return msg.wParam ;
}

long FAR PASCAL WndProc (HWND hWnd, unsigned iMessage, WORD wParam,
    LONG lParam)
{
    static BOOL    bCapturing = FALSE, bBlocking = FALSE, bStarted = FALSE ;
    static POINT   beg, end, oldend ;
    static short   xSize, ySize ;
    static HANDLE  hInstance ;
    HDC            hDC, hMemDC ;
    BITMAP         bm ;
    HBITMAP        hBitmap ;
    HICON          hIcon ;
    PAINTSTRUCT    ps ;
    switch (iMessage)
    {
        case WM_CREATE:    /* get program instance when window is created */
            hInstance = GetWindowWord (hWnd, GWW_HINSTANCE) ;
            break ;
        case WM_COMMAND:   /* one of the menu items has been clicked */
            switch (wParam)
            {
            case IDM_START:             /* the start capture item */
                bCapturing = TRUE ;
                bBlocking = bStarted = FALSE ;
                SetCapture (hWnd) ;                     /* grab mouse */
                SetCursor (LoadCursor (NULL, IDC_CROSS)) ;
                CloseWindow (hWnd) ;                    /* minimize window */
                break ;
            case IDM_CLEAR:     /* clears screen and clipboard */
                OpenClipboard (hWnd) ;
                EmptyClipboard () ;
                CloseClipboard () ;
                InvalidateRect (hWnd, NULL, TRUE) ;     /* forces paint */
                break ;
            case IDM_ABOUT:     /* show about box */
                MessageBox (hWnd, "Snap3 - Windows screen capture to clipboard.
                                                        \nJim Conger 1990.",
                    "Snap3 About", MB_OK) ;
                break ;
            case IDM_HELP:
                MessageBox (hWnd, "After you click the Start Capture menu
                                item, move the mouse to the upper left
                                of the area you want to copy to the
                                clipboard. Hold down the left mouse
                                button while you drag the mouse to the
                                lower right of the area. Once you release
                                the mouse button, the area is sent to the
                                clipboard and shown in Snap3's window.",
                    "Snap3 Help", MB_OK) ;
                break ;
            }
        case WM_LBUTTONDOWN:    /* starting capturing screen */
            if (bCapturing)
            {
                if (bStarted)
                {
                    bBlocking = TRUE ;
                    oldend = beg = MAKEPOINT (lParam) ;
                    OutlineBlock (hWnd, beg, oldend) ;
                    SetCursor (LoadCursor (NULL, IDC_CROSS)) ;
                }
                else
                    bStarted = TRUE ;
            }

            break ;
        case WM_MOUSEMOVE:      /* show area as rectangle on screen */
            if (bBlocking)
            {
                end = MAKEPOINT (lParam) ;
                OutlineBlock (hWnd, beg, oldend) ;  /* erase old outline */
                OutlineBlock (hWnd, beg, end) ;     /* draw new one */
                oldend = end ;
            }
            break ;
        case WM_LBUTTONUP:      /* capture and send to clipboard */
            if (bBlocking)
            {
                bBlocking = bCapturing = FALSE ;
                SetCursor (LoadCursor (NULL, IDC_ARROW)) ;
                ReleaseCapture () ;                 /* free mouse */

                end = MAKEPOINT (lParam) ;
                OutlineBlock (hWnd, beg, oldend) ;  /* erase area outline */
                xSize = abs (beg.x - end.x) ;
                ySize = abs (beg.y - end.y) ;
                hDC = GetDC (hWnd) ;
                hMemDC = CreateCompatibleDC (hDC) ;
                hBitmap = CreateCompatibleBitmap (hDC, xSize, ySize) ;

                if (hBitmap)
                {
                    SelectObject (hMemDC, hBitmap) ;
                    StretchBlt (hMemDC, 0, 0, xSize, ySize,
                        hDC, beg.x, beg.y, end.x - beg.x,
                        end.y - beg.y, SRCCOPY) ;
                    OpenClipboard (hWnd) ;
                    EmptyClipboard () ;
                    SetClipboardData (CF_BITMAP, hBitmap) ; /* copy to */
                    CloseClipboard () ;                     /* clipboard */
                    InvalidateRect (hWnd, NULL, TRUE) ;     /* request paint*/
                }
                else
                    MessageBeep (0) ;
                DeleteDC (hMemDC) ;
                ReleaseDC (hWnd, hDC) ;
            }
            ShowWindow (hWnd, SW_RESTORE) ;     /* un-minimize window */
            break ;
        case WM_PAINT:      /* display contents of clipboard if bitmap */
            hDC = BeginPaint (hWnd, &ps) ;
            if (IsIconic (hWnd))    /* if window is iconic, show icon */
            {
                hIcon = LoadIcon (hInstance, "snap3") ;
                if (hIcon != NULL)
                    DrawIcon (hDC, 1, 1, hIcon) ;
            }
            else                    /* if not, show clipboard contents */
            {
                OpenClipboard (hWnd) ;
                if (hBitmap = GetClipboardData (CF_BITMAP)) /* if bitmap */
                {
                    hMemDC = CreateCompatibleDC (hDC) ;
                    SelectObject (hMemDC, hBitmap) ;
                    GetObject (hBitmap, sizeof (BITMAP), (LPSTR) &bm) ;
                    SetStretchBltMode (hDC, COLORONCOLOR) ;
                    StretchBlt (hDC, 0, 0, xSize, ySize, hMemDC, 0, 0,
                        bm.bmWidth, bm.bmHeight, SRCCOPY) ;
                    DeleteDC (hMemDC) ;
                }
                CloseClipboard () ;
            }

            EndPaint (hWnd, &ps) ;
            break ;
        case WM_DESTROY:
            PostQuitMessage (0) ;
            break ;
        default:
            return DefWindowProc (hWnd, iMessage, wParam, lParam) ;
        }
    return 0L ;
}

/* OutlineBlock() writes a rectangle on screen given two corner points. R2_NOT
   style is used, so drawing twice on the same location erases outline. */
void OutlineBlock (HWND hWnd, POINT beg, POINT end)
{
    HDC   hDC ;

    hDC = CreateDC ("DISPLAY", NULL, NULL, NULL) ;
    ClientToScreen (hWnd, &beg) ;       /* convert to screen units */
    ClientToScreen (hWnd, &end) ;
    SetROP2 (hDC, R2_NOT) ;             /* use logical NOT brush */
    MoveTo (hDC, beg.x, beg.y) ;        /* draw rectangle */
    LineTo (hDC, end.x, beg.y) ;
    LineTo (hDC, end.x, end.y) ;
    LineTo (hDC, beg.x, end.y) ;
    LineTo (hDC, beg.x, beg.y) ;
    DeleteDC (hDC) ;
}
```